Having thus described the invention, it is now claimed:

1. A system for encrypting data, comprising:

a memory for storing permutating data values for decryption;

a bit table for tracking data modifications within the memory; and

a controller for executing an encryption algorithm wherein a plurality of data values are read from the memory generally simultaneously to determine a plurality of index values, and a plurality of pairs of data values stored in the memory are respectively swapped within the memory generally simultaneously, said plurality of pairs of data values identified by said plurality of index values.

2. The system of claim 1 wherein the memory is a dual port RAM memory for allowing simultaneous read and write operations.

3. The system of claim 1 wherein the memory is a single port RAM memory.

4. The system of claim 1 wherein said controller includes an implementation for detecting when said plurality of pairs of data values have been modified.

5. The system of claim 1 wherein the bit table comprises one bit per location in the memory.

6. The system of claim 1 further comprising a key memory for storing in consecutive order a plurality of keys respectively associated with a plurality of data frames including encrypted data, wherein each said key is comprised of a plurality of key values.

7. The system of claim 6 wherein the key memory comprises a dual port RAM memory.

8. A system for decrypting data, comprising:

a memory for storing permutating data values for decryption;

a bit table for tracking data modifications within the memory; and

a controller for executing a decryption algorithm wherein a plurality of data values are read from the memory generally simultaneously to determine a plurality of index values, and a plurality of pairs of data values stored in the memory are respectively swapped within the memory generally simultaneously, said plurality of pairs of data values identified by said plurality of index values.

9. The system of claim 8 wherein the memory is a dual port RAM memory for allowing simultaneous read and write operations.

10. The system of claim 8 wherein the memory is a single port RAM memory.

11. The system of claim 8 wherein said controller includes an implementation for detecting when said plurality of pairs of data values have been modified.

12. The system of claim 8 wherein the bit table comprises one bit per location in the memory.

13. The system of claim 8 further comprising a key memory for storing in consecutive order a plurality of keys respectively associated with a plurality of data

frames including encrypted data, wherein each said key is comprised of a plurality of key values.

14. The system of claim 13 wherein the key memory comprises a dual port RAM memory.

15. A method for encrypting data, comprising:

storing permutating data values for encryption;

tracking data modifications during the step of storing permutating values; and

executing an encryption algorithm wherein a plurality of data values are read from the stored permutating data values generally simultaneously to determine a plurality of index values, storing a plurality of pairs of data values, and respectively swapping generally simultaneously, said plurality of pairs of data values identified by said plurality of index values.

16. The method of claim 15 comprising a step of detecting when said plurality of pairs of data values have been modified.

17. The method of claim 15 comprising a step of forwarding the stored permutating data values when said data values have common data storage locations to correctly compute an out of order sequence of data manipulation during a same clock cycle.

18. The method of claim 15 where read/write operations between different algorithm iterations are mapped to different ports on a data memory in the same clock cycle.

19. The method of claim 15 of examining the stored data values to see if a simultaneous read/write operation is required.

20. The method of claim 15 further comprising the step of storing in consecutive order a plurality of keys respectively associated with a plurality of data frames including encrypted data, wherein each said key is comprised of a plurality of key values.

21. The method of claim 15 comprising a step of detecting when said plurality of pairs of data values have been modified.

22. A method for decrypting data, comprising:

storing permutating data values for decryption;

tracking data modifications during the step of storing permutating values; and

executing a decryption algorithm wherein a plurality of data values are read from the stored permutating data values generally simultaneously to determine a plurality of index values, storing a plurality of pairs of data values and respectively swapping generally simultaneously, said plurality of pairs of data values identified by said plurality of index values.

23. The method of claim 22 comprising a step of detecting when said plurality of pairs of data values have been modified.

24. The method of claim 22 comprising a step of forwarding the stored permutating data values when said data values have common data storage locations to correctly compute an out of order sequence of data manipulation during a same clock cycle.

25. The method of claim 22 where read/write operations between different algorithm iterations are mapped to different ports on a data memory in the same clock cycle.

26. The method of claim 22 of examining the stored data values to see if a simultaneous read/write operation is required.

27. The method of claim 22 further comprising the step of storing in consecutive order a plurality of keys respectively associated with a plurality of data frames including encrypted data, wherein each said key is comprised of a plurality of key values.

28. The method of claim 22 comprising a step of detecting when said plurality of pairs of data values have been modified.

29. A method of tracking changes to data locations in a memory comprising:

providing a bit table having at least one bit corresponding to an initial configuration of at least one respective addressed data location in a memory;

marking the at least one bit to track changes in the memory, wherein the step of marking includes changing the state of the at least one bit from the initial configuration, to correspond to a respective change in the respective addressed data location;

presuming an initial configuration of the addressed data location in the event the state of the at least one bit is not changed;

reading the addressed data location in the event the state of the at least one bit is changed.

30. The method of claim 29 wherein the least one bit comprises a plurality of bits corresponding to a respective plurality of addressed data locations, and wherein:

the step of presuming comprises presuming an initial configuration of only the addressed data locations where the state of the respective bits are not changed; and wherein

the step of reading comprises reading only the addressed data locations where the state of the respective bits are changed.

31. The method of claim 30 wherein the plurality of bits comprises 256 bits corresponding to a respective number of addressed data locations in a memory.

32. The method of claim 29 further comprising a step of clearing the at least one bit to an initial configuration corresponding to a respective initial state of the respective at least one respective addressed data location.

33. A bit table for tracking changes in data locations comprising:

at least one bit corresponding to at least one respective addressed data location in a memory;

a setting implementation for clearing the at least one bit to an initial configuration corresponding to a respective initial state of the respective at least one respective addressed data location;

a marking implementation for tracking changes in the memory, wherein the marking implementation changes the state of the at least one bit from the initial configuration, to correspond to a respective change in the respective addressed data

location, wherein an initial configuration of the addressed data location is presumed unless a respective corresponding change in the bit table is indicated.

34. The bit table of claim 33 wherein the at least one bit comprises 256 bits corresponding to a respective number of addressed data locations in a memory.

35. The bit table of claim 33 wherein the at least one bit comprises a flip flop.